

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ М. В. ЛОМОНОСОВА  
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

## ОТЧЕТ ПО ЗАДАНИЮ №2

Выполнили:  
Данько Артем  
Ковалева Василина  
Шматенко Дарья

Москва  
2019

# Содержание

Постановка задачи	2
Ход решения	3
Описание программы	3
Необходимые компоненты	6
Участники	6

## Постановка задачи

После оглушительного успеха в освобождении Астапора, Миэрина и Юнка от власти работорговцев Дейенерис Бурерожденная открыла себе доступ к Летнему морю, а следовательно – путь в Вестерос.

Для ведения войны с Семью Королевствами нужно оружие, а для оружия нужна сталь. Нет никаких сомнений в кузнечном искусстве Безупречных, однако поставщики стали не столь надежны.

Два основных поставщика стали – это Westeros Inc. и Harpy & Co. На протяжении нескольких месяцев мы покупаем сталь у обеих компаний, и каждая из них предлагает ощутимую скидку при заключении эксклюзивного договора на поставку.

Советник королевы Тирион Ланнистер знает о твоём умении принимать взвешенные рациональные решения и просит помощи в объективном решении вопроса о том, с какой из компаний следует заключить эксклюзивный договор на поставку стали.

У Тириона есть записи о производстве мечей каждым из кузнецов-безупречных, а также данные о количестве сломанных мечей в каждый из месяцев ведения боевых действий.

Данные имеют вид:

	<i>unsullen.id</i>	<i>production.date</i>	<i>report.date</i>	<i>produced</i>	<i>defects</i>	<i>supplier</i>
0	1.0	1	1	103.0	0.0	harpy.co
1	1.0	1	2	0	2.0	harpy.co
2	1.0	1	3	0	4.0	harpy.co
3	1.0	1	4	0	5.0	harpy.co
4	1.0	1	5	0	13.0	harpy.co

## Ход решения

Мы получили данные о качестве стали компаний Westeros Inc. и Harpy&Co, построив различные графики:

1. Общее количество произведенного и сломавшегося оружия для каждой из компаний за все месяцы сотрудничества
2. Количество произведенного и сломавшегося оружия обеих компаний для каждого месяца производства
3. Среднее число поломанной продукции для каждого месяца
4. Среднее число поломанной продукции по сроку службы
5. Качество продукции за каждый месяц (процент сломавшейся продукции, которая была произведена в  $i$ -й месяц)
6. Зависимость количества поломок от кузнеца

## Описание программы

1. `all_time(book)`

Посмотрим для каждого поставщика, сколько всего мечей было произведено и какое количество из них сломалось. Используя `drop`, исключим колонки `report.date`, `production.date`, `unsullen.id` за ненадобностью. С помощью `groupby` сгруппируем данные по поставщику и применим `sum`, таким образом, для каждой компании получим суммарное количество произведённой продукции и число сломавшихся мечей. Сгенерируем объекты `fig` и `axes`. Используя `subplots`, будем строить диаграммы для каждого поставщика. Перебираем компании, вызывая функцию `book.supplier.unique()`, и формируем с помощью `loc` подтаблицы, где хранится информация только про одного поставщика. Теперь с помощью `plot` отобразим на диаграммах данные для каждого производителя с номером `number`, где по оси ординат будут отложены `produced` и `defects`, а две колонки будут отвечать за количество произведённых и сломавшихся мечей. Положение диаграмм зададим используя `axes` и `number`.

2. `crack_and_make(book)`

Используя `drop`, исключим колонки `report.date`, `unsullen.id` за ненадобностью. С помощью `groupby` сгруппируем данные по поставщику и месяцу изготовления, а затем применим `sum`. Таким образом получим суммарное количество произведённой продукции в каждый месяц и число сломавшихся мечей. Создадим новые наборы данных `prod_book` и `crack_book`. Для этого с помощью `pivot` преобразуем таблицу, используя значения столбцов или индексов. Индексом будет служить `production.date`, колонками – `supplier`. Для набора `prod_book` ячейки будут заполнены значениями из колонки `produced`, для `crack_book` – `defects`. Сгенерируем объекты `fig` и

axes, используя subplots, и построим две диаграммы для произведённых и сломавшихся мечей, с помощью plot и 'ax' задав их положение.

### 3. month\_cracked\_prod(book)

Определим для каждого production.date среднюю долю сломанных к данному месяцу мечей. Используя drop, исключим колонки report.date, unsullen.id за ненадобностью. С помощью groupby сгруппируем данные по поставщику и месяцу изготовления, далее применим sum, таким образом получим суммарное количество произведённой продукции в каждый месяц и число сломавшихся мечей. Поделим число поломок на число произведённых изделий, запишем полученное частное в колонку defects и разделим её на число месяцев (7 - book['production.date']), прошедших со дня изготовления продукции. Построим график. Перебирая поставщиков (book.supplier.unique()), формируем с помощью loc подтаблицы, где хранится информация только про одного поставщика. Теперь отобразим данные на графике с помощью fig.plot, где по оси ординат отложены defects, а по оси абсцисс – месяц изготовления.

### 4. mean\_after\_month(book)

Определим среднее число сломавшихся мечей после каждого месяца эксплуатации, то есть среднее количество поломок в месяц после одного месяца использования, после двух и т.д. Вычтем из report.date колонку production.date и переименуем её с помощью rename в time\_to\_death, получая срок службы. Используя drop, исключим колонки unsullen.id, produced за ненадобностью. С помощью groupby сгруппируем данные по месяцу изготовления, поставщику и сроку службы, следом применим sum. После, исключив столбец production.date, перегруппируем по поставщику и сроку службы. Вслед за этим, используя mean, возьмём среднее и построим график. Перебираем поставщиков, вызывая функцию book.supplier.unique(), и формируем с помощью loc подтаблицы, где хранится информация только про одного поставщика. Теперь отобразим данные на графике, где по оси ординат отложены defects.

### 5. unsullens(book)

Проверим не портит ли какой-либо кузнец общий результат. Используя drop, исключим колонки report.date, production.date за ненадобностью. С помощью groupby сгруппируем данные по поставщику и кузнецу, а затем применим sum. Таким образом, получаем для каждого кузнеца суммарное количество произведённой им продукции в каждый месяц и число сломавшихся мечей. Построим график, далее сгенерируем объекты fig и axes, используя subplots. Также введём переменную number, отвечающую за номер поставщика. Перебираем поставщиков, вызывая функцию book.supplier.unique(), и формируем с помощью loc подтаблицы, где хранится информация только про одного производителя. Индексной делаем колонку, содержащую идентификационный номер кузнеца. Используя drop, исключим за ненадобностью колонки supplier, unsullen.id. Теперь, с помощью plot, отобразим на диаграммах данные для каждого производи-

теля с номером `number`, где по оси ординат будут отложены `produced` и `defects`, а по оси абсцисс – номер кузнеца.

#### 6. `crack_production(book)`

Используя `drop`, исключим колонки `report.date`, `unsullen.id` за ненадобностью. С помощью `groupby` сгруппируем данные по поставщику и месяцу изготовления и применим `sum`. Таким образом, получаем суммарное количество произведённой продукции в каждый месяц. Подсчитаем сколько процентов из всей продукции, произведенной в *i*-й месяц, сложилось к текущему моменту, то есть найдём отношение `book['defects']` к `book['produced']` и сохраним его в колонку `percent`. Построим график. Перебираем поставщиков, вызывая функцию `book.supplier.unique()`, и формируем с помощью `loc` подтаблицы, где хранится информация только про одного поставщика. Теперь, отобразим данные на графике, где по оси ординат отложены `percent`.

## Необходимые компоненты

- Библиотеки и функции

- pandas – библиотека, предназначенная для хранения таблиц. Также содержит огромное количество универсальных функций для их комфортной обработки.
- copy – предоставляет общие операции копирования
- matplotlib – для работы с графиками
- read\_csv(filepath\_or\_buffer, sep=', ') – считывает csv файл в dataframe, sep – разделитель
- copy.deepcopy(x) – возвращает полную копию x
- drop – удаляет указанную строку или столбец
- groupby() – группирует элементы таблицы по колонке
- sum() – суммирует элементы группы
- rename(columns) – переименование колонок
- plt.figure(figsize=None).add\_subplot(numrows, numcols, fignum) – figsize задаёт размер, add\_subplot добавляет объект для рисования графика по указанным координатам
- unique() – возвращает уникальные значения объекта Series
- loc – доступ к группе строк и столбцов по меткам или логическому выражению
- fig.plot(x, y, color, label) – рисует график, где точки имеют координаты (x,y), label используется для генерации легенды
- plt.show() – показ изображения (графика)
- plt.subplots(nrows, ncols) – возвращает объекты Figure и Axes
- pivot(index, columns, values)  
index и columns – столбцы, используемые для создания индекса и столбца нового датафрейма  
values – столбцы, используемые для заполнения значений нового фрейма

- Программы

- Jupyter Notebook

## Участники

Данько Артем, Ковалева Василина, Шматенко Дарья.  
Все этапы работы были выполнены совместно.